Introduction to the ACTS Collection Why/How Do We Use These Tools?

Tony Drummond
Computational Research Division
Lawrence Berkeley National Laboratory

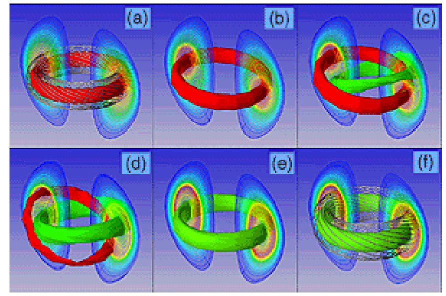
Sixth ACTS Collection Workshop
August 24-26, 2005 - Berkeley, CA 94703

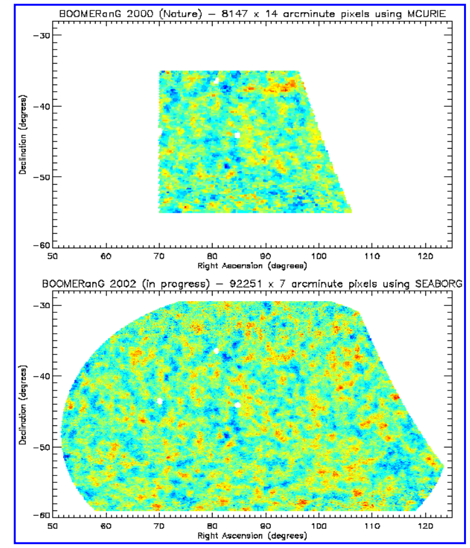
Introduction to The ACTS Collection

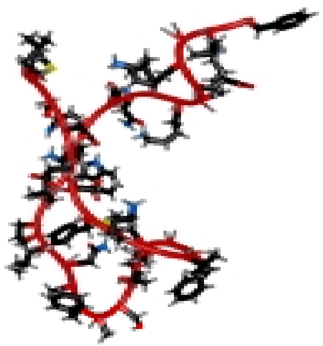
Where are the applications?

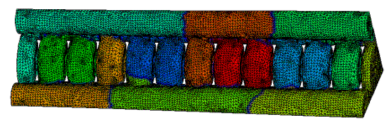
- Accelerator Science
- Astrophysics
- Biology
- Chemistry
- Earth Sciences
- Materials Science
- Nanoscience
- Plasma Science

:

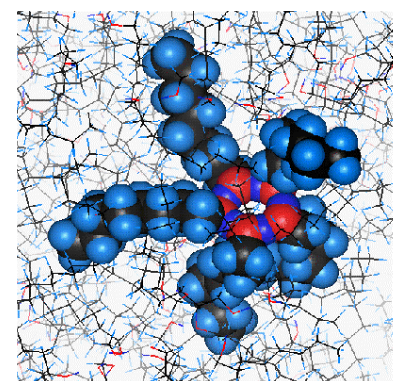








Omega3P is a parallel distributed-memory code intended for the modeling and analysis of accelerator cavities, which requires the solution of generalized eigenvalue problems. A parallel exact shift-invert eigensolver based on PARPACK and SuperLU has allowed for the solution of a problem of order 7.5 million with 304 million nonzeros.



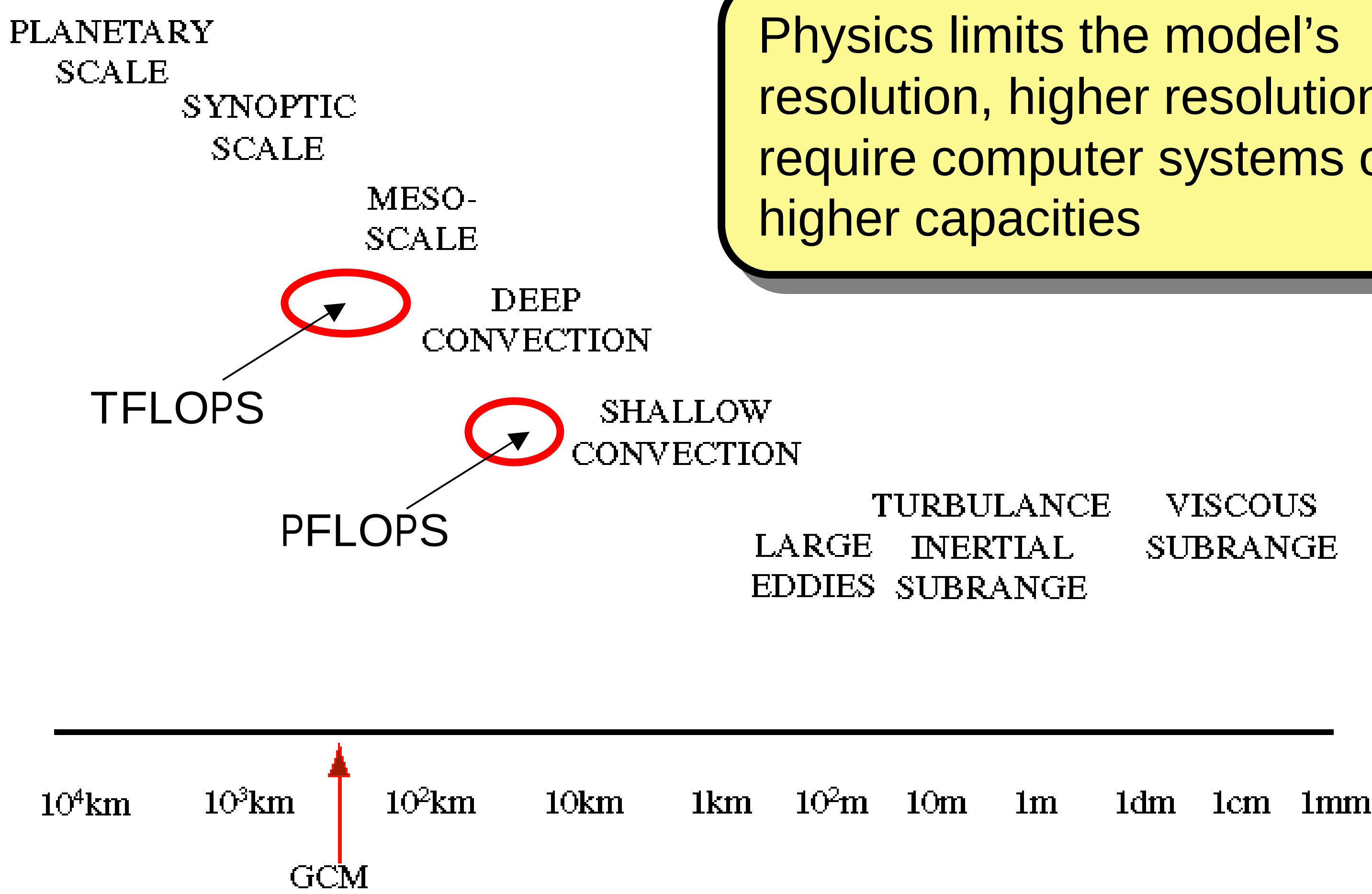
Commonalities:

- Major advancements in Science
- Increasing demands for computational power
- Rely on available computational systems, languages, and software tools

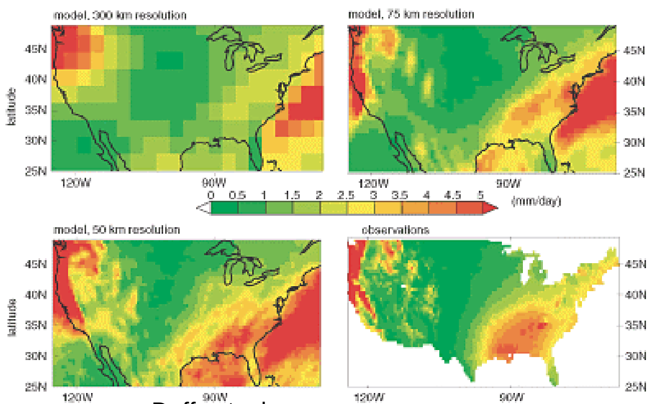
Increasing Computational Demand
(Area: Atmospheric Research)

Physics limits the model's resolution, higher resolution will require computer systems of higher capacities

SPECTRUM OF ATMOSPHERIC PHENOMENA

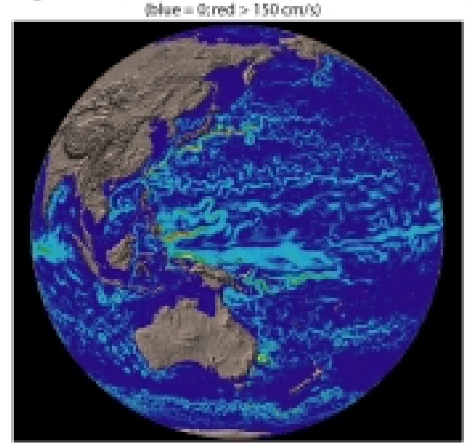


Multidisciplinary Research
(Area: Climate Research)



Duffy et. al.,
Lawrence Livermore National Laboratory

1/10 Degree Global POP Ocean Model Currents at 50m Depth
(blue = 0; red > 150 cm/s)



Mathew E. Maltruda and
Julie L. McClean

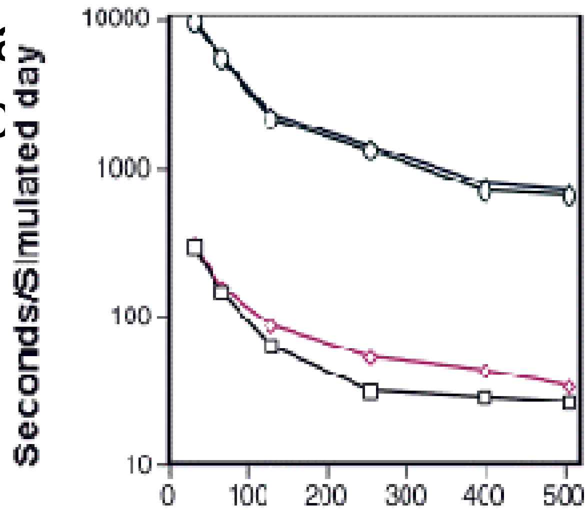
- Atmospheric general circulation model
  - Dynamics
  - Sub-grid scale parameterized physics processes
    - Turbulence, solar/infrared radiation transport, clouds.
- Oceanic general circulation model
  - Dynamics (mostly)
- Sea ice model
  - Viscous elastic plastic dynamics
  - Thermodynamics

- Land Model
  - Energy and moisture budgets
  - Biology
- Chemistry
  - Tracer advection, possibly stiff rate equations.
- Ocean Biology

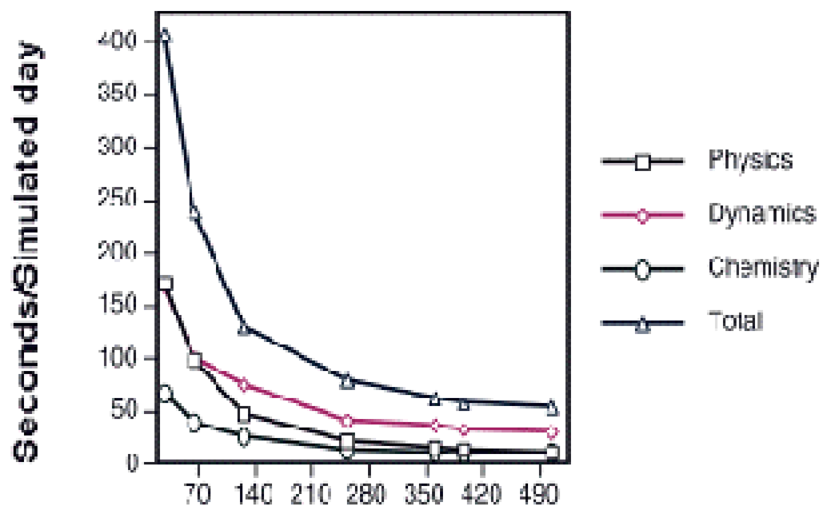
Some Computational Challenges *In Climate Research*

Climate Models:

- Higher resolutions are computational demanding
- No-trivial load-balancing
- Coupling different physics, times and spatial doma perfc



AGCM/ACM
2.5 long x 2 lat, 30 layers
25-chemical species



AGCM/ACM
2.5 long x 2 lat, 30 layers
2-chemical species

Key Lesson Learned

(Software Development)

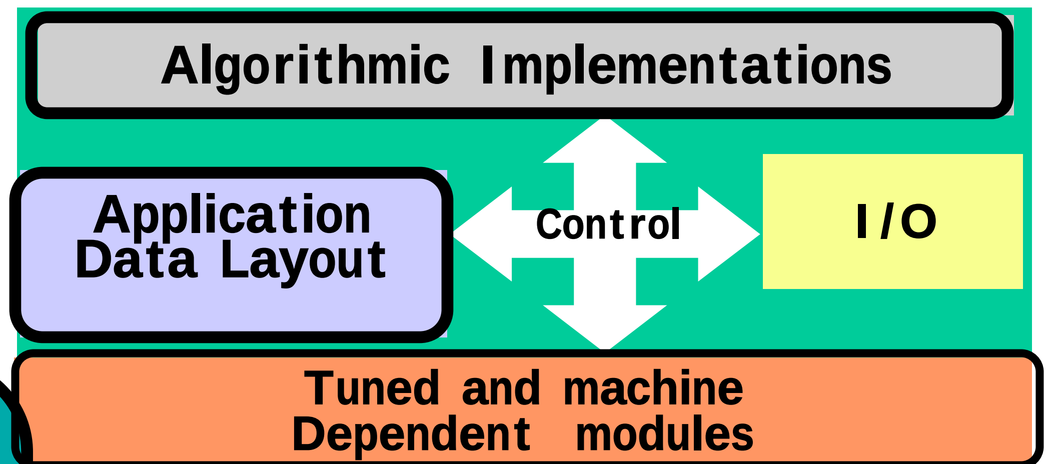
"We need to move away from a coding style suited for serial machines, where every macrostep of an algorithm needs to be thought about and explicitly coded, to a higher-level style, where the compiler and library tools take care of the details. And the remarkable thing is, if we adopt this higher-level approach right now, even on today's machines, we will see immediate benefits in our productivity."

W. H. Press and S. A. Teukolsky, 1997

Numerical Recipes: Does This Paradigm Have a future?

Key Lesson Learned
(Software Development)

Changes in algorithms sometimes lead to several years advancement in computations. Needs Flexibility!

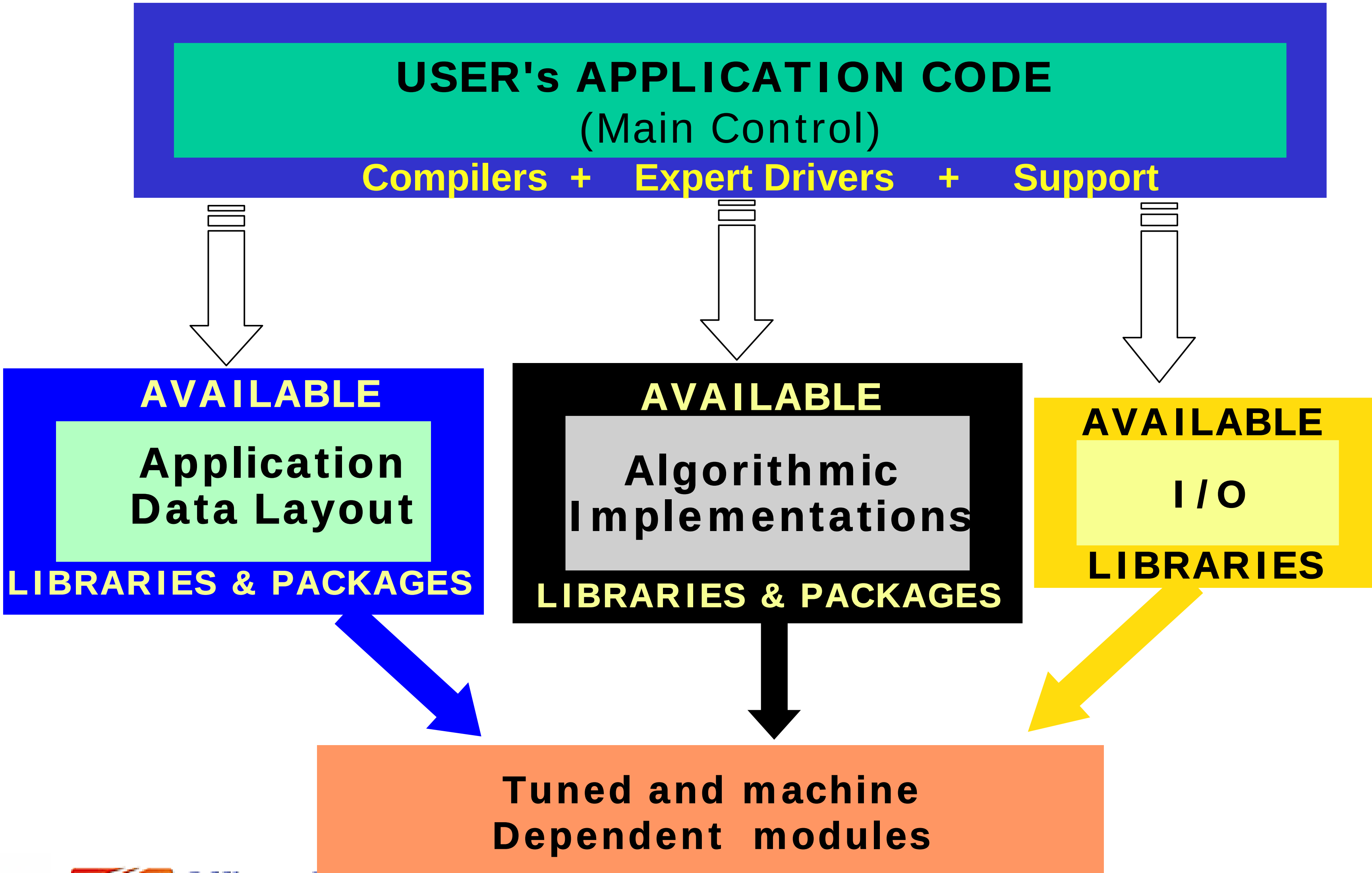


Its performance is influenced by system parameters and in steps in the algorithm. Critical points: portability and scalability.

New Architecture requires extensive tuning, may even require new programming paradigms. This is **Difficult to maintain and not "very" portable.**

Key Lesson Learned

(Software Development)



Lesson =
High Quality Software Reusability

- Scientific or engineering context
- Domain expertise

- Simulation codes
- Data Analysis codes

General Purpose Libraries

- Data Structures
- Algorithms
- Code Optimization
- Programming Languages
- O/S - Compilers

Hardware - Middleware - Firmware

Lesson = High Quality Software Reusability

Funded by DOE/ASCR

Library Development

Numerical Tools

Code Development

Run Time Support

http://acts.nersc.gov

General Purpose Libraries

- **Data Structures**
- **Algorithms**
- **Code Optimization**
- **Programming Languages**
- **O/S - Compilers**

Hardware - Middleware - Firmware

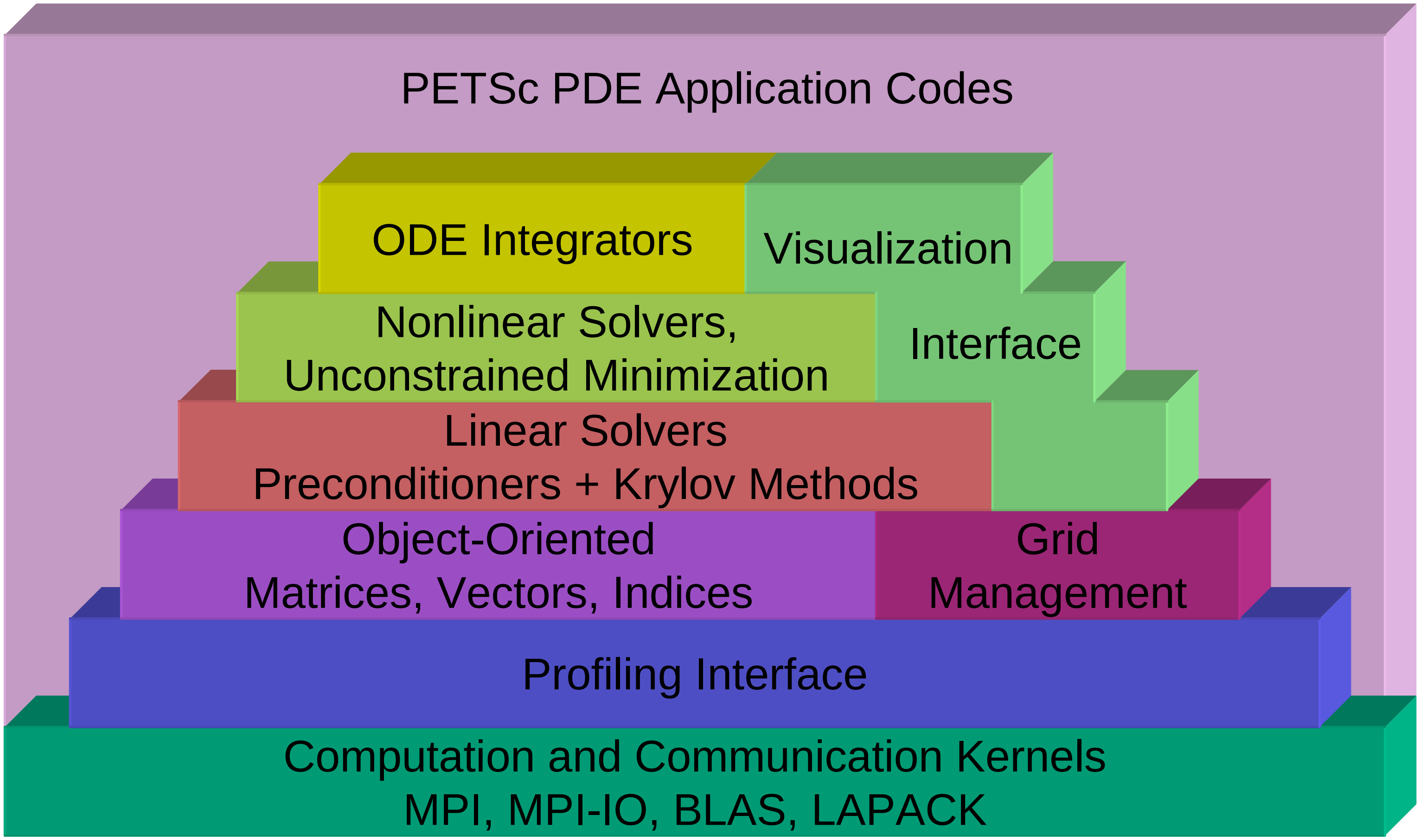
ACTS Numerical Tools: *Functionality*

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations	Direct Methods	LU Factorization	ScaLAPACK(dense) SuperLU (sparse)
		Cholesky Factorization	ScaLAPACK
		LDL^T (Tridiagonal matrices)	ScaLAPACK
		QR Factorization	ScaLAPACK
		QR with column pivoting	ScaLAPACK
		LQ factorization	ScaLAPACK

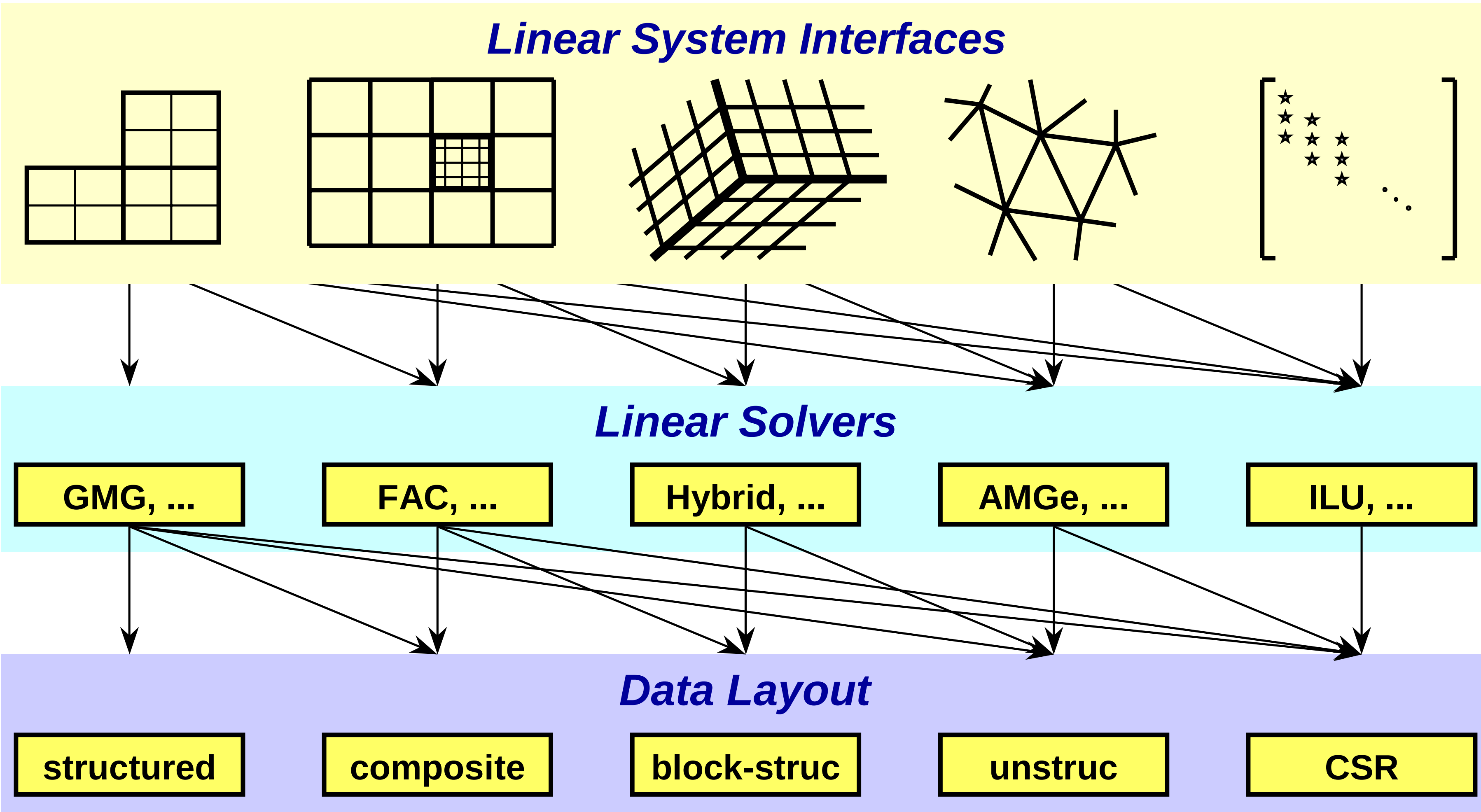
ACTS Numerical Tools: *Functionality*

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations (*cont*..)	Iterative Methods	Conjugate Gradient	AztecOO (Trilinos) PETSc
		GMRES	AztecOO PETSc Hypre
		CG Squared	AztecOO PETSc
		Bi-CG Stab	AztecOO PETSc
		Quasi-Minimal Residual (QMR)	AztecOO
		Transpose Free QMR	AztecOO PETSc

Structure of PETSc



Hypre Conceptual Interfaces



INTERFACE TO SOLVERS

List of Solvers and Preconditioners per Conceptual Interface

Solvers	System Interfaces			
	Struct	SStruct	FEI	IJ
Jacobi	X			
SMG	X			
PFMG	X			
BoomerAMG	X	X	X	X
ParaSails	X	X	X	X
PILUT	X	X	X	X
Euclid	X	X	X	X
PCG	X	X	X	X
GMRES	X	X	X	X

ACTS Numerical Tools: *Functionality*

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations (*cont*..)	Iterative Methods (*cont*..)	SYMMLQ	PETSc
		Precondition CG	AztecOO PETSc Hypre
		Richardson	PETSc
		Block Jacobi Preconditioner	AztecOO PETSc Hypre
		Point Jocobi Preconditioner	AztecOO
		Least Squares Polynomials	PETSc

ACTS Numerical Tools: *Functionality*

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations (*cont*..)	Iterative Methods (*cont*..)	SOR Preconditioning	PETSc
		Overlapping Additive Schwartz	PETSc
		Approximate Inverse	Hypre
		Sparse LU preconditioner	AztecOO PETSc Hypre
		Incomplete LU (ILU) preconditioner	AztecOO
		Least Squares Polynomials	PETSc
	MultiGrid (MG) Methods	MG Preconditioner	PETSc Hypre
		Algebraic MG	Hypre
		Semi-coarsening	Hypre

ACTS Numerical Tools: *Functionality*

Computational Problem	Methodology	Algorithm	Library
Linear Least Squares Problems	Least Squares	$\min_x \| b - Ax \|_2$	ScaLAPACK
	Minimum Norm Solution	$\min_x \| x \|_2$	ScaLAPACK
	Minimum Norm Least Squares	$\min_x \| b - Ax \|_2$ $\min_x \| x \|_2$	ScaLAPACK
Standard Eigenvalue Problem	Symmetric Eigenvalue Problem	$Az = \lambda z$ For $A=A^H$ or $A=A^T$	ScaLAPACK (dense) SLEPc (sparse)
Singular Value Problem	Singular Value Decomposition	$A = U\Sigma V^T$ $A = U\Sigma V^H$	ScaLAPACK (dense) SLEPc (sparse)
Generalized Symmetric Definite Eigenproblem	Eigenproblem	$Az = \lambda Bz$ $ABz = \lambda z$ $BAz = \lambda z$	ScaLAPACK (dense) SLEPc (sparse)

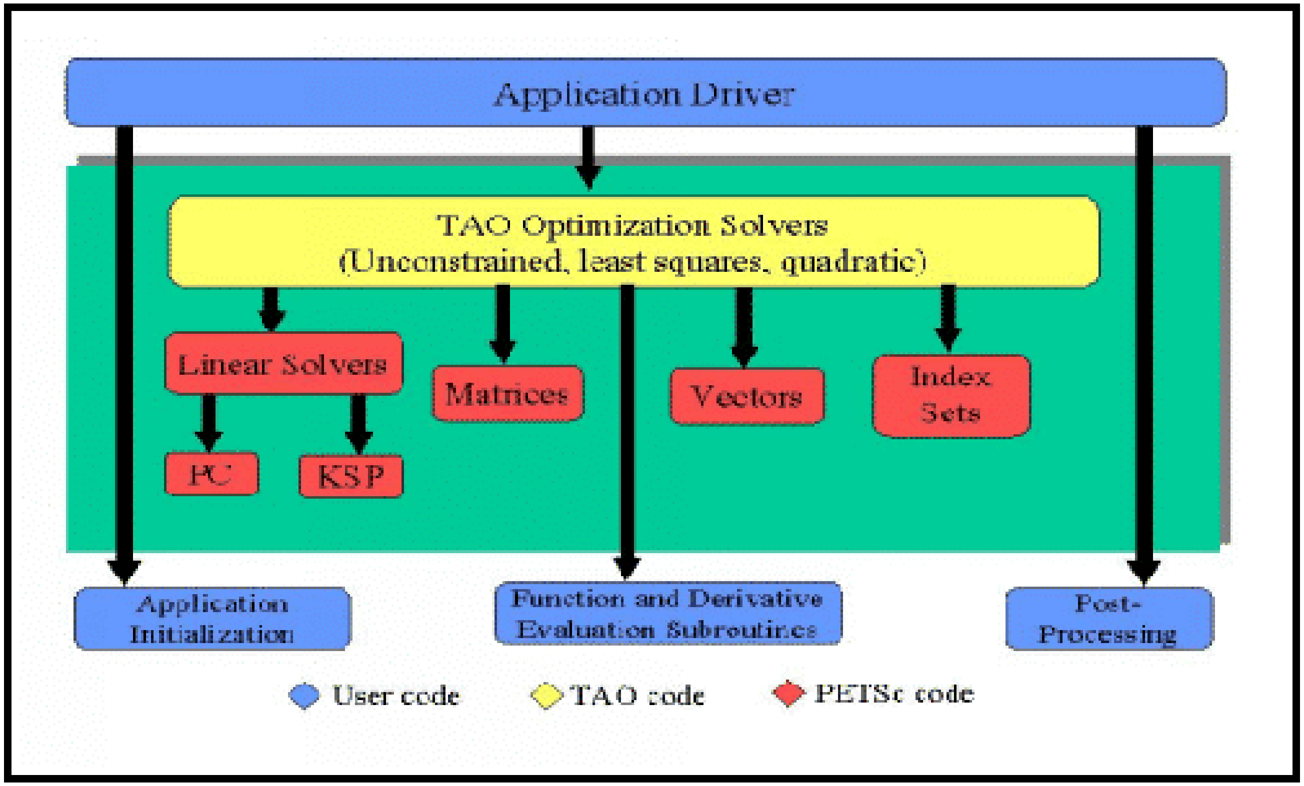
ACTS Numerical Tools: *Functionality*

Computational Problem	Methodology	Algorithm	Library
Non-Linear Equations	Newton Based	Line Search	PETSc
		Trust Regions	PETSc
		Pseudo-Transient Continuation	PETSc
		Matrix Free	PETSc

ACTS Numerical Tools: *Functionality*

Computational Problem	Methodology	Algorithm	Library
Non-Linear Optimization	Newton Based	Newton	OPT++ TAO
		Finite-Difference Newton	OPT++ TAO
		Quasi-Newton	OPT++ TAO
		Non-linear Interior Point	OPT++ TAO
	CG	Standard Non-linear CG	OPT++ TAO
		Limited Memory BFGS	OPT++
		Gradient Projections	TAO
	Direct Search	No derivate information	OPT++

TAO - Interface with PETSc



OPT++ Interfaces

- Four major classes of problems available
  - *NLF0(ndim, fcn, init_fcn, constraint)*
    - Basic nonlinear function, no derivative information available
  - *NLF1(ndim, fcn, init_fcn, constraint)*
    - Nonlinear function, first derivative information available
  - *FDNLF1(ndim, fcn, init_fcn, constraint)*
    - Nonlinear function, first derivative information approximated
  - *NLF2(ndim, fcn, init_fcn, constraint)*
    - Nonlinear function, first and second derivative information available

ACTS Numerical Tools: *Functionality*

Computational Problem	Methodology	Algorithm	Library
Non-Linear Optimization	Newton Based	Newton	OPT++ TAO
		Finite-Difference Newton	OPT++ TAO
		Quasi-Newton	OPT++ TAO
		Non-linear Interior Point	OPT++ TAO
	CG	Standard Non-linear CG	OPT++ TAO
		Limited Memory BFGS	OPT++
		Gradient Projections	TAO
	Direct Search	No derivate information	OPT++

ACTS Numerical Tools: *Functionality*

Computational Problem	Methodology	Algorithm	Library
Non-Linear Optimization (cont..)	Semismoothing	Feasible Semismooth	TAO
		Unfeasible semismooth	TAO
Ordinary Differential Equations	Integration	Adam-Moulton (Variable coefficient forms)	CVODE (SUNDIALS) CVODES
	Backward Differential Formula	Direct and Iterative Solvers	CVODE CVODES
Nonlinear Algebraic Equations	Inexact Newton	Line Search	KINSOL (SUNDIALS)
Differential Algebraic Equations	Backward Differential Formula	Direct and Iterative Solvers	IDA (SUNDIALS)

ACTS Numerical Tools: *Functionality*

Computational Problem	Support	Techniques	Library
Writing Parallel Programs	Distributed Arrays	Shared-Memory	Global Arrays
		Distributed Memory	CUMULVS (viz) Globus (Grid)
		Grid Generation	OVERTURE
		Structured Meshes	CHOMBO (AMR) Hypre OVERTURE PETSc
		Semi-Structured Meshes	CHOMBO (AMR) Hypre OVERTURE
	Distributed Computing	GRID	Globus
		Remote Steering	CUMULVS
		Coupling	PAWS

ACTS Numerical Tools: *Functionality*

Computational Problem	Support	Technique	Library
Writing Parallel Programs (cont.)	Distributed Computing	Check-point/restart	CUMULVS
Profiling	Algorithmic Performance	Automatic instrumentation	PETSc
		User Instrumentation	PETSc
	Execution Performance	Automatic Instrumentation	TAU
		User Instrumentation	TAU
Code Optimization	Library Installation	Linear Algebra Tuning	ATLAS
Interoperability	Code Generation	Language	BABEL CHASM
		Components	CCA

Software Reusability

What have we gained? What are the goals?

min[*time_to_first_solution*] (prototype)

min[*time_to_solution*] (production)

- Outlive Complexity
  - Increasingly sophisticated models
  - Model coupling
  - Interdisciplinary

(Software Evolution)

- Sustained Performance
  - Increasingly complex algorithms
  - Increasingly diverse architectures
  - Increasingly demanding applications

(Long-term deliverables)

min[*software-development-cost*]

max[*software_life*] and **max**[*resource_utilization*]

Minimum Requirements for Reusable High Quality Software Tools

- **Robustness**
  - Maintained across platforms
  - Compiler independent
  - Precision Independent
  - Error Handling
  - Check Pointing

Minimum Requirements for Reusable High Quality Software Tools

- Robust
- **Scalable (across large Petascale systems)**

Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- **Extensible (New Algorithms, New Techniques)**

Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- Extensible
- **Interoperable**
  - Frameworks/PSE
  - Tool-to-Tool
  - **Component Technology**
    - **More Flexible**
    - **Retains better Robustness, Scalability, and Extensibility**
    - **Long term pay-offs**

http://www.cca-forum.org

Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- Extensible
- Interoperable
- **User Friendly Interfaces**
- **Well documented**

User Interfaces

```
CALL BLACS_GET( -1, 0, ICTXT )
 CALL BLACS_GRIDINIT( ICTXT, 'Row-major', NPROW, NPCOL )
:
 CALL BLACS_GRIDINFO( ICTXT, NPROW, NPCOL, MYROW, MYCOL )
:
:
 CALL PDGESV( N, NRHS, A, IA, JA, DESCA, IPIV, B, IB, JB, DESCB,
 $              INFO )
```

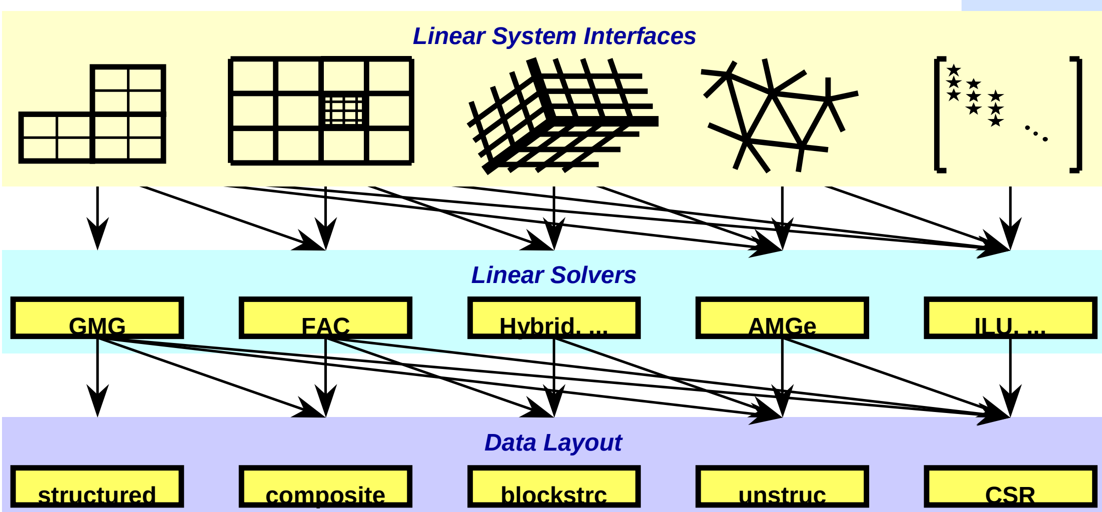
Library Calls

- -ksp_type [cg,gmres,bcgs,tfqmr,…]
- -pc_type [lu,ilu,jacobi,sor,asm,…]

More advanced:

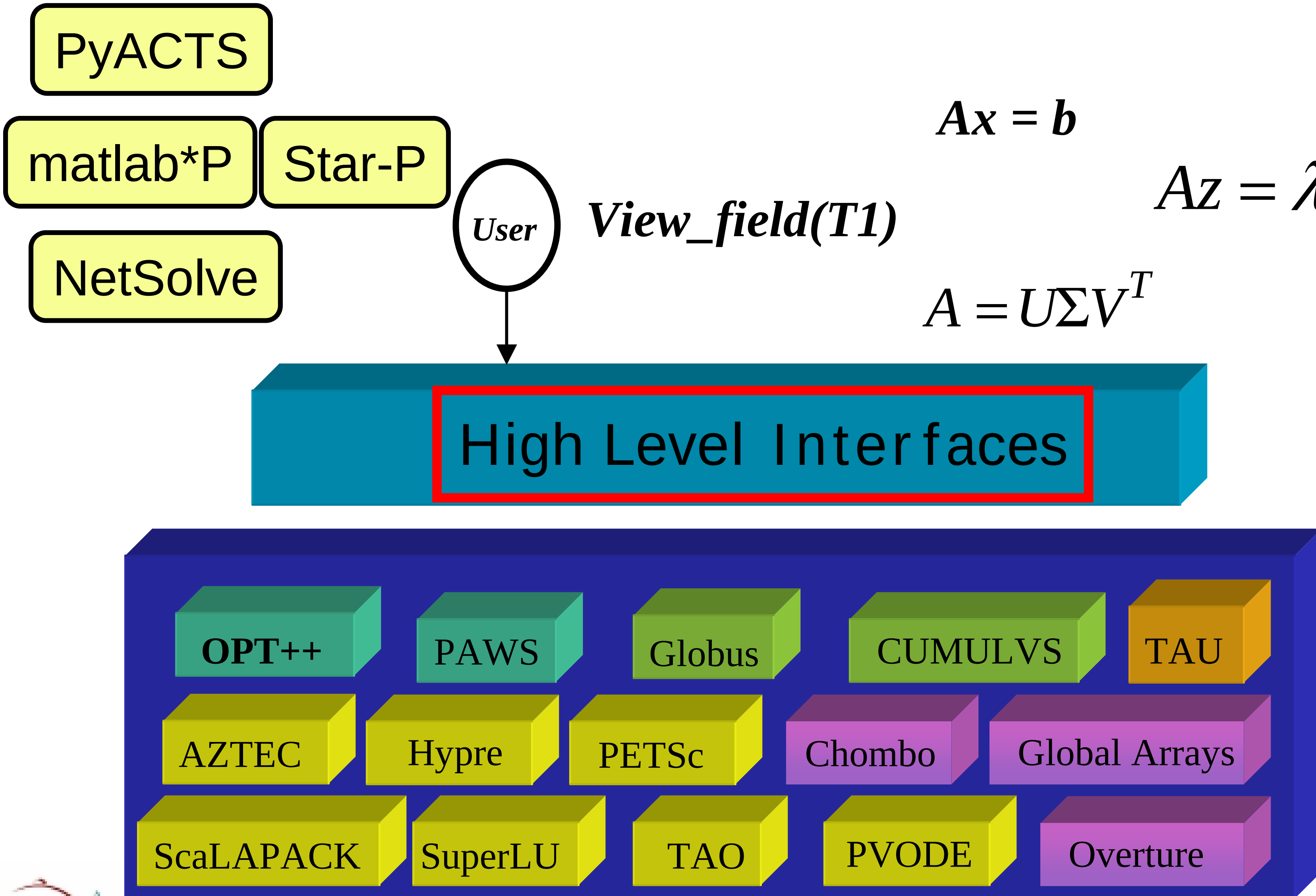
- -ksp_max_it <max_iters>
- -ksp_gmres_restart <restart>
- -pc_asm_overlap <overlap>

Command lines



Problem Domain

User Interfaces



Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- Extensible
- Interoperable
- User Friendly Interfaces
- Well documented
- **Periodic Tests and Evaluations**

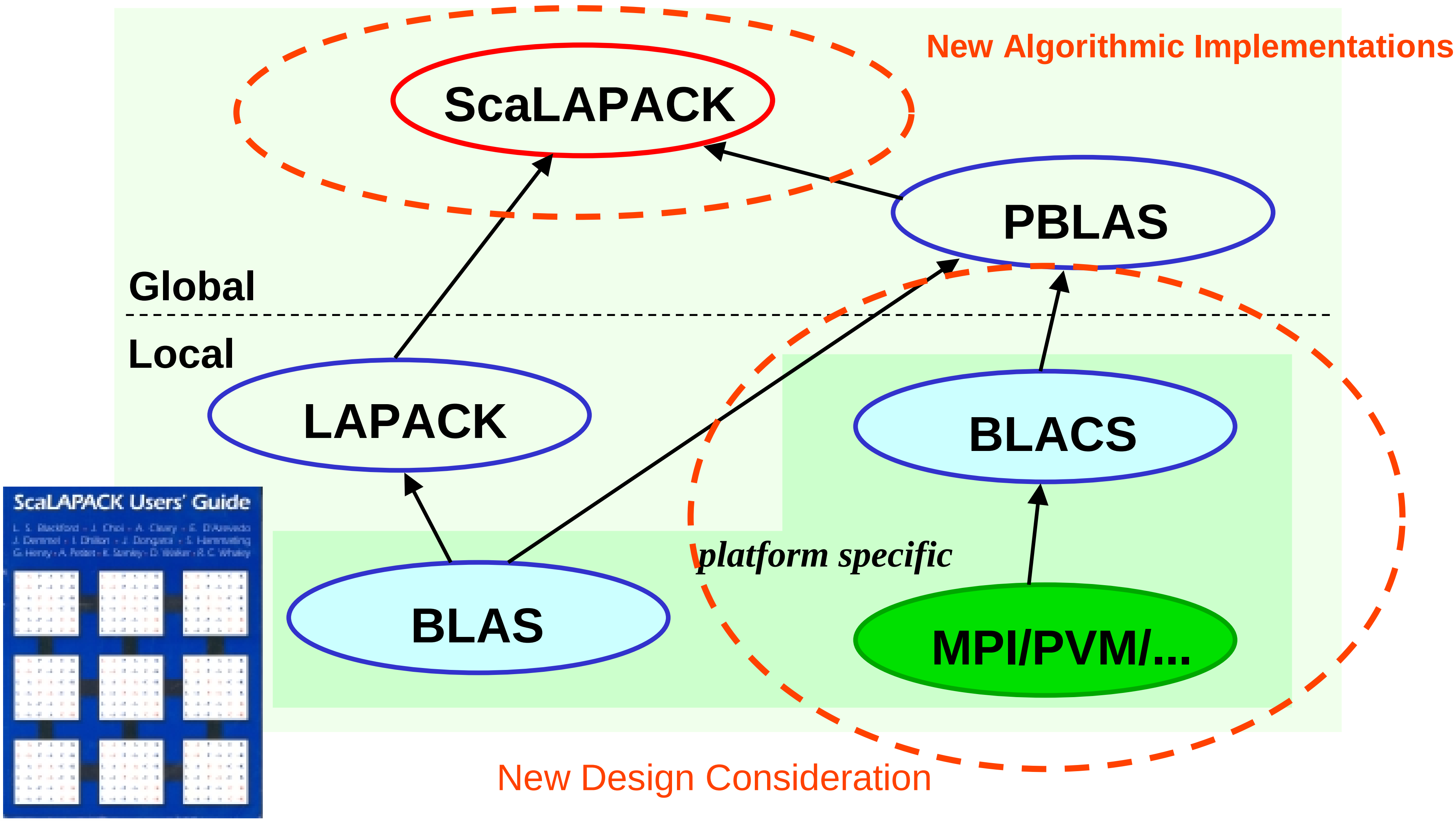
Versions (tools, systems, O/S, compilers)

- Sanity-check (robustness)
- Interoperability (maintained)
- Consistent Documentation

Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- Extensible
- Interoperable
- User Friendly Interfaces
- Well documented
- Periodic Tests and Evaluations
- **Portability and Fast Adaptability (The Evolution)**

Tool Evolution
Example: ScaLAPACK



Acknowledgments

- Department of Computer Science, Indiana University, for facilitating the use of their cluster for the GA, TAU and CCA hands-on tutorials
- National Energy Research Scientific Computing Center (NERSC) for the use of their IBM SP (seaborg) for the ScaLAPACK, SuperLU, PETSc, and SLEPc hands-on tutorials
- Yeen Mankin for all the great management and support running all the logistics of the workshop
- Suzanne Stevenson for reproduction of all the workshop materials
- To all the speakers and teams represented at the Sixth DOE ACTS Collection Workshop

This week at the Workshop

Tuesday *Perseverance Hall*	Wednesday *Perseverance Hall*	Thursday *Perseverance Hall*	Friday *Perseverance Hall*
Breakfast 8:00 – 8:30	**Breakfast** 8:00 – 8:30	**Breakfast** *8:00 – 8:30*	**Breakfast** 8:00 – 8:30
Welcome Remarks and Introduction **O. Marques** **T. Drummond** 08:30-10:00	**PDEs** *PETSC Part I* **M. Knepley** 08:30-10:00	**Tuning** *ATLAS* **T. Drummond** 08:30 –09:15	**PDEs** *MCT* **J. Larson** 08:30 -09:30
		PDE *Invited Talk* **R. Kirby** 09:15-10:00	**Steering** *CUMULVS* **J. Kohl** 09:30 – 10:30
Coffee Break 10:00 – 10:30	**Coffee Break** 10:00 – 10:30	**Coffee Break** 10:00 – 10:30	**Coffee Break** 10:30 – 11:00

Tuesday ***Perseverance Hall***	**Wednesday** ***Perseverance Hall***	**Thursday** ***Perseverance Hall***	**Friday** ***Perseverance Hall***
ScaLAPACK **O. Marques** 10:30 –11:30	*PETSc* *Part II* **M. Knepley** 10:30 –12:30	*SUNDIALS* **R. Serban** 10:30 –11:30	*CCA* **CCA-Team*** 11:00 –12:30
SuperLU **X. Li** 11:30-12:30		*Overture* **B. Henshaw** 11:30 –12:30	
Lunch 12:30-13:30	**Lunch** 12:30-13:30	**Lunch** 12:30-13:30	**Lunch** 12:30 – 13:30
Hypre **R. Falgout** 13:30 –14:30	*SLEPc* **J. Roman** 13:30 –14:30	*Global Arrays* **B. Palmer** **M. Krishnan** 13:30 –14:30	*CCA* **CCA-Team*** 13:30 –15:00
Trilinos **M.Heroux** 14:30 –15:30	*TOPS* **D. Keyes** 14:30 –15:30	*TAU* **S. Shende** 14:30 –15:30	**Coffee Break** 15:00 – 15:30
Coffee Break 15:30 – 16:00	**Coffee Break** 15:30 – 16:00	**Coffee Break** 15:30 – 16:00	***212 Wheeler UCB***
212 Wheeler UCB	***212 Wheeler UCB***	***212 Wheeler UCB***	*CCA hands-On* **CCA Team*** 15:30 – 18:30
SuperLU *Hands-On* **X. LI** 16:00 – 17:00	*PETSc* *Hands-on* **M. Knepley** 16:00 – 17:00	*GA Hands-On* **B. Palmer** **M. Krishnan** 16:00 – 17:00	
ScaLAPACK *Hands-On* **O. Marques** 17:00 – 18:00	*SLEPc* *Hands-On* **J. Roman** 17:00 – 18:00	*TAU* *Hands-On* **S. Shende** 17:00 – 18:00	
Dinner I- House ***H.D. Simon*** *7:00 – 9:00*			

This week at the Workshop

Hands-On:

- Your login name should be written in your badge
- Passwords:

On Tolman Hall PCs: **c@1summer**

On NERSC computers:

acts04*bbxx* , where

bb = Your initials (first name and last name) in lowercase

xx = Your training account number, this can be one or two digits

Examples:

1. Joe Lone, user id train55 , password = **acts05jl55**
2. Mary Lee Second, user id train1, password = **acts05ml1**

This week at the Workshop

- **Picture today at lunch break outside the cafeteria!**
- **Return your Vouchers for travel!!**
- **Return your sign policy forms**

THANK YOU